Universidade do Minho
Escola de Engenharia

Departamento de Informática

Licenciatura em Engenharia Informática

Laboratórios de Informática III

Projeto de Laboratórios de Informática III – 1.ª Fase

Grupo n.º 24

Alexandre Rodrigues Fernandes, n.º A94154

Orlando José da Cunha Palmeira, n.º A97755

Rui Pedro Guise da Silva, n.º A97133

Braga, novembro de 2021

Índice

1 Introdução

Esta primeira fase do projeto é dividida em dois exercícios nos quais nos foi proposto que implementássemos, na linguagem de programação *C*, um programa que recolhe dados (referentes a utilizadores, commits e repositórios) provenientes de ficheiros CSV e que os valida do seguinte modo:

1. O processo é iniciado através da validação dos dados dos ficheiros pelo seu formato, isto é, se as informações constantes em cada linha do ficheiro estão corretamente escritas. Por exemplo, se o identificador de um utilizador é um número inteiro positivo ou se uma data está dentro dos critérios definidos para este projeto (formato AAAA-MM-DD hh:mm:ss, posterior a 7 de abril de 2005).
2. Após a primeira validação (ponto 1), os dados de cada ficheiro serão validados conforme a sua compatibilidade com outros ficheiros, isto é, se existem commits referentes a utilizadores e repositórios existentes e se existem repositórios referentes a utilizadores existentes e que possuam commits associados.

2 Resolução dos problemas e estratégias adotadas

Para o primeiro exercício a validação de cada um dos ficheiros de input é feita através de uma função, específica para cada um deles, que avalia individualmente os parâmetros de uma dada linha do ficheiro, fazendo recurso a funções auxiliares.
De modo a obter cada um dos campos a serem avaliados é primeiramente necessário percorrer a linha em questão até ao caracter ';', visto que este é responsável pela delimitação do início/final de cada parâmetro. Posteriormente, a sub-string resultante do processo anterior será então avaliada de acordo com as limitações impostas ao parâmetro que esta representa (por exemplo inteiro não negativo, data, etc.).
Numa tentativa de melhorar o desempenho deste exercício, estabeleceu-se que as funções responsáveis pela validação dos campos devolvessem o valor 0, se o mesmo não cumprisse as restrições impostas, ou caso contrário, o comprimento da sub-string associada ao parâmetro. A ação anterior tem como objetivo a obtenção do número de posições que é necessário avançar na linha, de modo que esta aponte para a primeira posição do próximo parâmetro a avaliar, deixando assim de ser necessário a utilização de um ciclo que procure o próximo caracter ';'.

Para o segundo exercício, recorreu-se à implementação de árvores binárias de procura para o armazenamento de indentificadores (ID's) de utilizadores, visto que este ficheiro se manterá inalterado nesta etapa do projeto, e repositórios, para comparação futura. Na criação das referidas árvores foi necessária a leitura do respetivo identificador, presente em cada uma das linhas, assim como a sua inserção na mesma, seguindo a ordem presente no ficheiro.
Na porção do código responsável pela remoção de commits remetentes a utilizadores/repositórios inválidos foi criada uma função que recolhe os ID's (author_id, repo_id, commiter_id) dos mesmos e, posteriormente, procede à sua procura nas árvores correspondentes. No caso de um dado commit se apresentar válido (ou seja, o seu autor e repositório são também válidos) será então também criada uma nova árvore binária, cujo conteúdo será composto pelos ID's dos repositórios que se encontrem não vazios.
Finalmente, na secção responsável pela remoção de repositórios inválidos realizam-se as procuras do ID dos seus criadores, assim como a dos seus próprios identificadores nas suas respetivas árvores. Seguindo todos os passos expostos acima, ficam então cumpridos os objetivos propostos no presente guião.

3 Conclusão

Finda esta fase do projeto, verificámos que há alguns detalhes que poderiam ser melhorados. Em primeiro lugar, no uso das árvores binárias, poderíamos ter utilizado a biblioteca glib que já contém implementações de funções de manipulação dessas estruturas. Contudo, após várias tentativas sem sucesso de instalar essa biblioteca, acabámos por implementar as nossas próprias ferramentas para poder concluir a entrega no tempo imposto.
Seguidamente, também achamos que poderíamos equilibrar as árvores em tempo de execução de modo a aumentar a performance da procura. No entanto, também constatámos que estar sempre a equilibrar a árvore sempre que é feita uma inserção poderia ter um impacto acentuado na eficiência do programa pelo que optámos por mantê-la na sua forma original.

De um modo geral, achámos que o nosso trabalho está bastante satisfatório uma vez que, após vários testes, os objetivos foram cumpridos.